

# Symplectic integrator for $s$ -dependent static magnetic fields based on mixed-variable generating functions

---

**A. Wolski<sup>a,\*</sup>, J. Gratus<sup>b,c</sup> and R. Tucker<sup>b,c</sup>**

<sup>a</sup>*Department of Physics, University of Liverpool,  
Liverpool, UK*

<sup>b</sup>*Physics Department, Lancaster University,  
Bailrigg, Lancaster, UK*

<sup>c</sup>*The Cockcroft Institute,*

*Daresbury, Warrington, UK*

*E-mail: a.wolski@liverpool.ac.uk*

**ABSTRACT:** We describe a technique for constructing a symplectic transfer map for a charged particle moving through an accelerator component with arbitrary three-dimensional static magnetic field. The transfer map is constructed by symplectic integration; by representing the map at each step of the integration by a mixed-variable generating function, exact symplecticity is ensured. By using an appropriate integration algorithm, there is no necessity to make the paraxial approximation. The technique is illustrated by application (in one degree of freedom) to a quadrupole magnet with strong octupole component and fringe field.

**KEYWORDS:** Accelerator modelling and simulations (multi-particle dynamics; single-particle dynamics); Beam dynamics.

---

\*Corresponding author.

---

## Contents

<b>1. Introduction</b>	<b>1</b>
<b>2. Symplectic integrators</b>	<b>3</b>
2.1 Integrator for the Hamiltonian in the paraxial approximation	4
2.2 Symplectic Runge-Kutta integrator	5
2.3 Generating function composition: induction method	7
2.4 Generating function composition: iteration method	9
<b>3. Example: quadrupole with octupole component and fringe field</b>	<b>10</b>
<b>A. Proof of the inductive formula for solving a system of polynomial equations</b>	<b>13</b>
<b>B. Generalised gradients</b>	<b>15</b>

---

## 1. Introduction

The ability to compute accurately and efficiently the trajectory of a particle in an accelerator beam line is fundamental to accelerator design and operation. Where the beam line includes components with complex electromagnetic fields, and where fast tracking and a high degree of accuracy are required, this problem presents significant challenges. Considerable work over the years has led to the development of a variety of techniques for particle tracking (see, for example, [1 – 5]); however, since the equations describing particle trajectories cannot be solved exactly, various approximations are always needed. Often, there is a compromise that must be made between accuracy and speed.

There are two approaches to tracking a particle through the electromagnetic field in a given accelerator component: either, the equations of motion may be integrated numerically, or, a transfer map may be applied for the entire component. Although the equations of motion are well known, numerical integration is in general too slow for long-term particle tracking in a storage ring. A transfer map, consisting of a set of equations relating the particle position and momentum at the exit of the component to the position and momentum at the entrance, provides a much more efficient tracking method; however, the problem then is to construct with good accuracy the transfer map for a given component. One approach is to integrate the equations of motion using a differential algebra (DA) code [6], to produce the transfer map in the form of a multi-variable power series. Although a map in this form is very convenient for particle tracking, in general the power series must be truncated at some order in each step of the integration, to keep the number of terms manageable. As a consequence of the truncation, the final map will not be symplectic: this may have an impact on detailed aspects of the particle dynamics (for example, whether or not a particle remains in a storage ring over a large number of turns [7–9]). Techniques have been devised to construct

symplectic, finite-order Taylor maps (Cremona maps) that correspond to a symplectic Taylor map that has been truncated at some order [10, 11]. However, these methods are not especially easy to implement, and the consequences of modifying a map to make it symplectic are not always clear [4].

The need for efficient symplectic tracking tools has motivated the development of techniques for manipulating symplectic transfer maps (symplectomorphisms) using, for example, representations commonly known as Lie transformations [5, 12]. Venturini and Dragt have shown how to construct the transfer map for an accelerator component in the form of a Lie transformation, using detailed magnetic field data [13]. While Lie transformations provide a valuable analytical framework, it is not straightforward to implement the relevant formulae in an accelerator simulation code. As an alternative, we discuss here the construction of a mixed-variable generating function [14], in particular, to represent the dynamics in an accelerator component with some specified static inhomogeneous magnetic field. Mixed-variable generating functions are already used for particle tracking where symplecticity is important [7, 9, 15–21]. Besides ensuring symplecticity, mixed-variable generating functions also have potential advantages over Taylor maps, in reducing the number of coefficients needed to describe the map, and in improving the accuracy of a map for a given accelerator configuration that is obtained by interpolation between known maps representing a set of “reference” configurations [22, 23]. However, the common approach to using mixed-variable generating functions is first to construct truncated multi-variable Taylor expansions for the canonical variables (approximating the transfer map, usually for an entire turn of a storage ring) and then to construct the generating function that gives this map to some desired order. Such a process can be used to “symplectify” a truncated power series; but then the same concerns apply as in the case of construction of Cremona maps from truncated power series. A more direct approach, that we consider here, consists of working entirely *ab-initio* with mixed-variable generating functions. The advantage is an improvement in efficiency and accuracy that can be achieved by avoiding conversions between different (and, in some cases, non-symplectic) representations of the same map.

Although the manipulation of maps in the form of mixed-variable generating functions is not trivial, it is at least possible to obtain from the generating function an exact set of algebraic equations relating the initial and final values of the dynamical variables that preserves the symplectic structure. Unfortunately, these equations then need to be solved numerically for given initial conditions, but this can be done to any desired precision with reasonable efficiency, using an iterative technique such as Newton’s method. The symplecticity of the map can then be preserved, to machine precision.

The above arguments suggest that a technique for constructing the generating function for a given accelerator component with specified static magnetic field offers a potentially useful tool in particle tracking: we describe such a technique in Section 2. We see that relatively straightforward expressions can be written down for the mixed-variable generating function for each integration step in the case that the paraxial approximation can be made for the Hamiltonian (Section 2.1); slightly more complicated (though still very manageable) expressions can be used where a more accurate form of the Hamiltonian is needed (Section 2.2). A key step in the integration is the composition of two generating functions: this is discussed in some detail in Sections 2.3 and 2.4, where we present two methods for performing the composition. Then, in Section 3, we illustrate

the techniques developed in Section 2 by applying them to the case of a magnetic quadrupole with strong octupole component and fringe field.

## 2. Symplectic integrators

We shall consider the case of a charged particle moving through a static magnetic field  $\vec{B} = \nabla \times \vec{A}$ . Let  $s$  parameterise the reference trajectory, which by assumption is an orbit in the magnetic field. The canonical variables on the (extended) phase space are  $(\vec{x}, s, \vec{p}, p_s)$  where  $\vec{x} = (x, y, z)$  and  $\vec{p} = (p_x, p_y, p_z)$ .  $x$  and  $y$  denote the position of the particle in a plane perpendicular to the reference trajectory at  $s$ , which the particle crosses at time  $t$ ; and  $z = s/\beta_0 - ct$  where  $\beta_0$  is the speed (as a fraction of the speed of light) of a particle with a fixed reference (kinetic) momentum  $P_0$ . The transverse components of the canonical momentum are defined by:

$$p_x = \frac{\gamma(|\vec{\beta}|)m\dot{x} + qA_x(\vec{x}, s)}{P_0}, \quad (2.1a)$$

$$p_y = \frac{\gamma(|\vec{\beta}|)m\dot{y} + qA_y(\vec{x}, s)}{P_0}, \quad (2.1b)$$

where  $\gamma(|\vec{\beta}|)$  is the relativistic factor; the particle velocity is  $\vec{\beta}c = (\dot{x}, \dot{y}, \dot{s})$ ;  $m$  and  $q$  are the mass and charge of the particle; and the dot indicates a derivative with respect to time. The momentum conjugate to the longitudinal co-ordinate  $z$  is the energy deviation  $p_z$ , defined by:

$$p_z = \frac{E}{P_0 c} - \frac{1}{\beta_0}, \quad (2.2)$$

where  $E = \gamma(|\vec{\beta}|)mc^2$  is the kinetic energy of the particle. The reference momentum can be written as  $P_0 = \beta_0 \gamma_0 mc$ , where  $\gamma_0 = \gamma(|\beta_0|)$ . Note that third component of the particle co-ordinate vector  $\vec{x} = (x, y, z)$  actually specifies the time at which a particle arrives at a given longitudinal position within the magnetic field; so for a magnetostatic field,  $\vec{A}$  is independent of  $z$ .

It is convenient to work with the normalised vector potential, defined by:

$$\vec{a} = \frac{\vec{A}}{B\rho}, \quad (2.3)$$

where  $B\rho = P_0/q$  is the beam rigidity and the vector  $\vec{a}$  has Frenet components  $(a_x, a_y, a_s)$ .

In these variables, the Hamiltonian for particle motion in a static magnetic field with a straight reference trajectory is:

$$H = \frac{p_z}{\beta_0} - \sqrt{\left(\frac{1}{\beta_0} + p_z\right)^2 - (p_x - a_x)^2 - (p_y - a_y)^2} - \frac{1}{\beta_0^2 \gamma_0^2} - a_s + p_s, \quad (2.4)$$

where  $\gamma_0 = \gamma(\beta_0)$ . The independent variable is denoted by  $\sigma$ . Note that:

$$\frac{ds}{d\sigma} = \frac{\partial H}{\partial p_s} = 1, \quad (2.5)$$

and therefore:

$$s = \sigma + \sigma_0, \quad (2.6)$$

where  $\sigma_0$  is an (arbitrary) constant of integration.

The method will be illustrated for a particle that moves with  $y = p_y = p_z = 0$ , i.e. we consider only the evolution of the variables  $x, p_x$  and  $s$ . The generalisation to the full number of degrees of freedom is straightforward, and requires no significant new features. In addition, we assume that the particle is ultra-relativistic, so that we can take the limit  $\gamma_0 \rightarrow \infty$ . By making an appropriate gauge transformation, it is always possible to work in a gauge in which one component of the vector potential vanishes at all positions in space: we shall choose  $a_y = 0$ . The Hamiltonian is then:

$$H = -\sqrt{1 - (p_x - a_x)^2} - a_s + p_s. \quad (2.7)$$

Our assumption that  $y = p_y = 0$  as the particle moves through the field requires that:

$$\left. \frac{\partial a_x}{\partial y} \right|_{y=0} = \left. \frac{\partial a_s}{\partial y} \right|_{y=0} = 0. \quad (2.8)$$

This is a restriction on the field, but again the generalisation to include other cases requires no significant new features.

An accelerator component is described by specifying the components of the vector potential. Neglecting radiation and collective effects, the transfer map for a particle in a magnetostatic field is symplectic. Therefore, the values of the canonical variables at the exit of the accelerator component can be related, at least in principle, to the values of the variables at the entrance by a mixed-variable generating function. Following Goldstein's nomenclature [14], with  $(x_1, p_{x1})$  the entrance variables and  $(x_2, p_{x2})$  the exit variables, we write  $F(x_1, p_{x2})$  as a generating function of the second kind. Then, the entrance and exit variables are related by:

$$x_2 = \frac{\partial F}{\partial p_{x2}} \quad (2.9a)$$

$$p_{x1} = \frac{\partial F}{\partial x_1}, \quad (2.9b)$$

Our objective is to develop a procedure for constructing  $F(x_1, p_{x2})$  in a given static magnetic field.

## 2.1 Integrator for the Hamiltonian in the paraxial approximation

Assuming that the dynamical variables and normalised vector potential have small values, we can make the paraxial approximation for the Hamiltonian given by Eq. (2.7):

$$H \approx -1 + \frac{1}{2}(p_x - a_x)^2 - a_s + p_s. \quad (2.10)$$

In this case, a method for constructing a convenient symplectic integrator has been described by Wu, Forest and Robin [24]. A more general symplectic integrator, not requiring the paraxial approximation, is described in Section 2.2. However, where it is possible to make the paraxial approximation without losing the desired level of accuracy, the advantage of doing so is that integration can be performed more simply. Following [24], using the Hamiltonian in the paraxial approximation Eq. (2.10), the integration of the particle motion through a step  $\Delta\sigma$  (corresponding to a step  $\Delta s = \Delta\sigma$  in the field) can be expressed as a sequence of transformations, for example:

$$\mathcal{M} = \mathcal{M}_1 \circ \mathcal{M}_2 \circ \mathcal{M}_3 \circ \mathcal{M}_2^{-1} \circ \mathcal{M}_1, \quad (2.11)$$

where:

$$\mathcal{M}_1 s = s + \frac{1}{2}\Delta\sigma, \quad (2.12a)$$

$$\mathcal{M}_1 p_x = p_x + \frac{1}{2}\Delta\sigma \frac{\partial a_s}{\partial x}, \quad (2.12b)$$

$$\mathcal{M}_2 p_x = p_x - a_x, \quad (2.12c)$$

$$\mathcal{M}_2^{-1} p_x = p_x + a_x, \quad (2.12d)$$

$$\mathcal{M}_3 x = x + \Delta\sigma p_x. \quad (2.12e)$$

The effect of the transformations  $\mathcal{M}_1$ ,  $\mathcal{M}_2$ ,  $\mathcal{M}_3$ , on variables not explicitly shown is to leave them unchanged. These transformations provide a second-order integrator (the error is  $O(\Delta\sigma^3)$ ).

If the components of the vector potential  $\vec{a}$  are expressed as power series in  $x$  and  $s$ , then at this stage we can use a differential algebra code to compose the transformations first for each step through the field, and then for successive steps through the field. However, except for trivial cases, this will require truncation of the power series in order maintain a manageable number of terms in the power series, and the truncation will mean a loss of symplecticity. Furthermore, we know from the symplectic condition that many of the coefficients in the power series will be related to each other: this direct approach is therefore more computationally expensive than necessary.

Instead, we can write down mixed-variable generating functions corresponding to each of the individual transformations. Using generating functions of the second kind, it can be seen that generators for the above transformations may be written:

$$F_1(x_1, p_{x2}, s_1, p_{s2}) = x_1 p_{x2} - \frac{1}{2}\Delta\sigma a_s(x_1, s_1) + s_1 p_{s2} + \frac{1}{2}\Delta\sigma p_{s2}, \quad (2.13a)$$

$$F_2(x_1, p_{x2}, s_1, p_{s2}) = x_1 p_{x2} + \int_0^{x_1} a_x(x, s_1) dx + s_1 p_{s2}, \quad (2.13b)$$

$$F_3(x_1, p_{x2}, s_1, p_{s2}) = x_1 p_{x2} + \frac{1}{2}\Delta\sigma p_{x2}^2 + s_1 p_{s2}, \quad (2.13c)$$

where  $F_i$  is the generator for the map  $\mathcal{M}_i$ . The generator for  $\mathcal{M}_2^{-1}$  is:

$$F_2^{-1}(x_1, p_{x2}, s_1, p_{s2}) = x_1 p_{x2} - \int_0^{x_1} a_x(x, s_1) dx + s_1 p_{s2}. \quad (2.14)$$

To proceed, we need a way to compose the generating functions. We discuss two possible methods for this in Sections 2.3 and 2.4. However, we first discuss an integrator, expressed in terms of mixed variable generating functions, that can be used even in cases where the paraxial approximation is not valid.

## 2.2 Symplectic Runge-Kutta integrator

In the previous section, we wrote down expressions for mixed-variable generating functions for integration steps where the dynamics could be described by a Hamiltonian in the paraxial approximation. While the expressions obtained can be expressed rather simply, there are cases of interest where the paraxial approximation is not valid. For such cases, at the cost of rather more complicated formulae, we can use an alternative integration method, such as a symplectic Runge-Kutta algorithm.

For a Hamiltonian system, a symplectic Runge-Kutta method may be written [25]:

$$x_2 = x_1 + \Delta\sigma \sum_{i=1}^n b_i \frac{\partial H(x^i, p_x^i)}{\partial p_x}, \quad (2.15a)$$

$$p_{x2} = p_{x1} - \Delta\sigma \sum_{i=1}^n b_i \frac{\partial H(x^i, p_x^i)}{\partial x}, \quad (2.15b)$$

where the “intermediate” variables  $(x^i, p_x^i)$  are given by:

$$x^i = x_1 + \Delta\sigma \sum_{i=1}^n a_{ij} \frac{\partial H(x^j, p_x^j)}{\partial p_x}, \quad (2.16a)$$

$$p_x^i = p_{x1} - \Delta\sigma \sum_{j=1}^n a_{ij} \frac{\partial H(x^j, p_x^j)}{\partial x}. \quad (2.16b)$$

The transformations (2.15) and (2.16) are symplectic if:

$$b_i a_{ij} + b_j a_{ji} = b_i b_j, \quad (2.17)$$

for all  $i$  and  $j$ . A simple case is given by  $n = 1$ ,  $a_{11} = 1/2$  and  $b_1 = 1$ : this provides a second-order symplectic integrator. A fourth-order integrator is given [26] by  $n = 2$ , and:

$$(a_{ij}) = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} - \frac{\sqrt{3}}{6} \\ \frac{1}{4} + \frac{\sqrt{3}}{6} & \frac{1}{4} \end{pmatrix}, \quad (2.18a)$$

$$(b_i) = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \end{pmatrix}. \quad (2.18b)$$

The transformations in Eq. (2.15) can be derived from a mixed-variable generating function of the second kind, given by [25]:

$$F = x_1 p_{x2} + \Delta\sigma \sum_{i=1}^n b_i H(x^i, p_x^i) - \Delta\sigma^2 \sum_{i,j=1}^n b_i a_{ij} \frac{\partial H(x^j, p_x^j)}{\partial x} \frac{\partial H(x^i, p_x^i)}{\partial p_x}. \quad (2.19)$$

To write the generating function explicitly, it is necessary first to solve for the intermediate variables  $(x^i, p_x^i)$  using Eq. (2.16), then use Eq. (2.15b) to eliminate  $p_{x1}$  (thus expressing the generating function  $F$  purely in terms of  $x_1$  and  $p_{x2}$ ).

Although each integration step requires the solution of a number of equations in several variables, it turns out that the solution can be accomplished in a straightforward fashion (using an iterative technique) with an algebraic code (such as Mathematica), or a DA code. We have implemented the fourth-order Runge-Kutta integrator (with coefficients (2.18)), using the mixed-variable generating function (2.19) in Mathematica; we return to the question of solving the systems of equations necessary to construct the mixed-variable generating function (in particular, Eqs. (2.16)) in Section 2.4.

Note that Wu, Forest, and Robin [24] discuss the extension of their method for the paraxial case, to cases where the paraxial approximation is not valid. However, this requires (local) time integration, which adds complexity to the problem. The advantage of the Runge-Kutta integrator is that one can adopt a unified approach, using (effectively) path length as the independent variable, for Hamiltonians of sufficiently general form for most particle tracking applications in accelerators.

### 2.3 Generating function composition: induction method

Let  $F_A(x_1, p_{x1})$  and  $F_B(x_2, p_{x2})$  be generating functions of the second representing the transformations from  $(x_1, p_{x1})$  to  $(x_2, p_{x2})$  and from  $(x_2, p_{x2})$  to  $(x_3, p_{x3})$  respectively. Our goal is to find a single mixed variable generating function  $F_C(x_1, p_{x3})$  that represents the effect of the combined transformation of  $F_A$  and  $F_B$ . We first note that  $F_A(x_1, p_{x2})$  and  $F_B(x_2, p_{x3})$  satisfy [14]:

$$\frac{dF_A}{d\sigma} = p_{x1}\dot{x}_1 + x_2\dot{p}_{x2}, \quad (2.20)$$

and

$$\frac{dF_B}{d\sigma} = p_{x2}\dot{x}_2 + x_3\dot{p}_{x3}. \quad (2.21)$$

where the dot now signifies differentiation with respect to the independent variable,  $\sigma$ . If we define:

$$F_C = F_A - x_2 p_{x2} + F_B, \quad (2.22)$$

we see that  $F_C$  satisfies:

$$\frac{dF_C}{d\sigma} = p_{x1}\dot{x}_1 + x_3\dot{p}_{x3}. \quad (2.23)$$

Therefore, if we express  $F_C$  as a function of  $x_1$  and  $p_{x3}$ , then  $F_C$  will provide a mixed variable generating function of the second kind, for a transformation from  $(x_1, p_{x1})$  to  $(x_3, p_{x3})$ . In general, Eq. (2.22) gives  $F_C$  as a function of  $x_1$ ,  $p_{x2}$ ,  $x_2$ , and  $p_{x3}$ . However we can use the equations:

$$x_2 = \frac{\partial F_A}{\partial p_{x2}}, \quad (2.24)$$

and

$$p_{x2} = \frac{\partial F_B}{\partial x_2}, \quad (2.25)$$

to eliminate  $p_{x2}$  and  $x_2$ . We are then left with  $F_C$  as a function of  $x_1$  and  $p_{x3}$  only, as required for a mixed-variable generating function of the second kind. If  $F_A$  and  $F_B$  are written as power series in their respective variables, then elimination of  $p_{x2}$  and  $x_2$  may be performed as follows. Equations (2.24) and (2.25) may be written in the form:

$$\begin{cases} \mathcal{P}_1(x_2, p_{x2}) = 0, \\ \mathcal{P}_2(x_2, p_{x2}) = 0, \end{cases} \quad (2.26)$$

where  $\mathcal{P}_1$  and  $\mathcal{P}_2$  are polynomials of order  $N$  in the given arguments. In general it is difficult to find exact solutions to (2.26). However, we only require perturbative solutions to some order in a perturbation parameter  $\varepsilon$ , where the dynamical variables are of order  $\varepsilon$ . Specifically,  $\varepsilon = 0$  implies  $(x_i, p_{xi}) = (0, 0)$  and we are on the ideal orbit. Thus we propose solving the perturbation equation:

$$\begin{cases} \mathcal{P}_1(x_2, p_{x2}) = O(\varepsilon^{N+1}), \\ \mathcal{P}_2(x_2, p_{x2}) = O(\varepsilon^{N+1}), \end{cases} \quad (2.27)$$

We define the vector  $\mathbf{X}$ :

$$\mathbf{X} = \begin{pmatrix} x_2 \\ p_{x2} \end{pmatrix}. \quad (2.28)$$



Then we can re-write Eq. (2.27) as:

$$\sum_{r=0}^N \hat{\mathbf{T}}_r(\overbrace{\mathbf{X}, \dots, \mathbf{X}}^r) = O(\varepsilon^{N+1}), \quad (2.29)$$

where each term in the summation is a vector, each component of which is a monomial of order  $r$  in the components of its vector arguments. Each of the  $\hat{\mathbf{T}}_r$  can be regarded as a tensor of rank  $r+1$ , so that  $\hat{\mathbf{T}}_0$  is a 2-component vector,  $\hat{\mathbf{T}}_1$  is a  $2 \times 2$  matrix, and so on. Using an index notation, we would write, for example:

$$[\hat{\mathbf{T}}_3(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3)]_i = \sum_{j,k,l=1}^2 [\hat{\mathbf{T}}_3]_{ijkl} [\mathbf{X}_1]_j [\mathbf{X}_2]_k [\mathbf{X}_3]_l. \quad (2.30)$$

The  $\hat{\mathbf{T}}_r$  are obtained from  $\mathcal{P}_1$  and  $\mathcal{P}_2$ , and are symmetric (i.e.  $[\hat{\mathbf{T}}_r(\mathbf{X}_1, \dots, \mathbf{X}_r)]_i$  is unchanged if any two of the arguments are interchanged).

We make two assumptions: first, that  $\lim_{\varepsilon \rightarrow 0}(\hat{\mathbf{T}}_1)$  is invertible; and second, that  $\hat{\mathbf{T}}_0 = O(\varepsilon)$ . Using the first assumption, we can define:

$$\mathbf{T}_k = -\hat{\mathbf{T}}_1^{-1} \hat{\mathbf{T}}_k. \quad (2.31)$$

In Appendix A, it is shown that a solution to Eq. (2.29) is:

$$\mathbf{X} = \sum_{r=1}^N \mathbf{c}_r, \quad (2.32)$$

where  $\mathbf{c}_1 = \mathbf{T}_0$ , and for  $r \geq 2$ :

$$\mathbf{c}_r = \sum_{k=2}^r \sum_{\substack{m_1, m_2, \dots, m_k=1 \\ m_1 + \dots + m_k = r}} \mathbf{T}_k(\mathbf{c}_{m_1}, \dots, \mathbf{c}_{m_k}). \quad (2.33)$$

Here the inner summation is performed over all combinations  $(m_1, m_2, \dots, m_k)$  where each  $m_i \geq 1$  and  $m_1 + \dots + m_k = r$ . Thus (2.33) can be rewritten as a multiple sum:

$$\mathbf{c}_r = \sum_{k=2}^r \sum_{m_1=1}^r \sum_{m_2=1}^{r-m_1} \sum_{m_3=1}^{r-m_1-m_2} \dots \sum_{m_{k-1}=1}^{r-m_1-\dots-m_{k-2}} \mathbf{T}_k(\mathbf{c}_{m_1}, \dots, \mathbf{c}_{m_{k-1}}, \mathbf{c}_{r-m_1-\dots-m_{k-1}}). \quad (2.34)$$

The  $\mathbf{c}_r$  are given inductively, so that:

$$\mathbf{c}_2 = \mathbf{T}_2(\mathbf{T}_0, \mathbf{T}_0), \quad (2.35)$$

$$\mathbf{c}_3 = \mathbf{T}_3(\mathbf{T}_0, \mathbf{T}_0, \mathbf{T}_0) + 2\mathbf{T}_2(\mathbf{T}_0, \mathbf{T}_2(\mathbf{T}_0, \mathbf{T}_0)), \quad (2.36)$$

and so on. We note that  $\mathbf{c}_r = O(\varepsilon^r)$ .

The above formulae may be generalised to any number of degrees of freedom. Thus, this procedure leads, in principle, to explicit expressions for the coefficients in  $F_C$  (expressed as a power series) in terms of the coefficients in  $F_A$  and  $F_B$ .

However, applied to the composition of generating functions, the solution expressed in equations (2.32) and (2.33) is not purely numerical, since the coefficients in the polynomial equations

(2.26) must be expressed in terms of dynamical variables (in the case of one degree of freedom,  $x_1$  and  $p_{x3}$ ). Unfortunately, for all but the simplest cases (low-order maps, in one degree of freedom), the expressions obtained are too complicated to be of real practical use. Even in one degree of freedom, the expressions up to fourth order are already very complicated. We therefore consider an iterative procedure for composing the generating functions, which simplifies the expressions required.

## 2.4 Generating function composition: iteration method

As an alternative to the induction method, we can attempt to solve the equations involved in generating function composition using iteration. The formulae required may be simpler to implement in practice. Suppose that we have two mixed variable generating functions,  $F_A(x_1, p_{x2})$  and  $F_B(x_2, p_{x3})$ , which are known.  $F_A$  gives the transformation from  $\sigma_1$  to  $\sigma_2$ :

$$x_2 = \frac{\partial F_A}{\partial p_{x2}}, \quad (2.37)$$

$$p_{x1} = \frac{\partial F_A}{\partial x_1}, \quad (2.38)$$

and  $F_B$  gives the transformation from  $\sigma_2$  to  $\sigma_3$ :

$$x_3 = \frac{\partial F_B}{\partial p_{x3}}, \quad (2.39)$$

$$p_{x2} = \frac{\partial F_B}{\partial x_2}. \quad (2.40)$$

We wish to find a generating function  $F_C(x_1, p_{x3})$  which gives the transformation from  $\sigma_1$  to  $\sigma_3$ :

$$x_3 = \frac{\partial F_C}{\partial p_{x3}}, \quad (2.41)$$

$$p_{x1} = \frac{\partial F_C}{\partial x_1}. \quad (2.42)$$

We have seen in Section 2.3 that if the generating functions are expressed as polynomials in the appropriate variables, then the problem can be reduced to one of finding solutions to a set of polynomial equations of the form (2.26). We can attempt to find a solution iteratively, using Newton's method, as follows. First, we construct the matrix  $J$ :

$$J = \begin{pmatrix} \frac{\partial \mathcal{P}_1}{\partial x_2} & \frac{\partial \mathcal{P}_1}{\partial p_{x2}} \\ \frac{\partial \mathcal{P}_2}{\partial x_2} & \frac{\partial \mathcal{P}_2}{\partial p_{x2}} \end{pmatrix}. \quad (2.43)$$

Then, if we make an initial estimate of the solution, for example:

$$\mathbf{X}_{(1)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad (2.44)$$

(where  $\mathbf{X}$  is given by Eq. (2.28)) then we can construct improved estimates:

$$\mathbf{X}_{(n+1)} = \mathbf{X}_{(n)} - J_{(n)}^{-1} \begin{pmatrix} \mathcal{P}_1 \\ \mathcal{P}_2 \end{pmatrix}_{(n)} \quad (2.45)$$

starting from  $n = 1$ . For Eq. (2.26), the solution needs to be performed using an algebraic code, since the polynomials are expressed in terms of more variables than just those for which a solution is sought. However, we find in practice (e.g. for the illustration presented in Section 3), that the iteration can be carried out quickly and converges rapidly, even when working to high order in the variables.

This approach to solving a system of nonlinear equations can also be applied to the Runge-Kutta integrator described in Section 2.2. If the Hamiltonian is expressed as a polynomial in the dynamical variables (note that this means expanding the square root in Eq. (2.7), but this can be done to higher order than is done for the paraxial approximation), then Eq. (2.16) represents  $2n$  polynomial equations in the  $2n$  “intermediate” variables  $(x^i, p_x^i)$ . These may be solved iteratively using Newton’s method (in an algebraic code), to express the intermediate variables as power series in the initial variables  $(x_1, p_{x1})$ . Then, it is only necessary to make a substitution of  $p_{x2}$  for  $p_{x1}$ , using Eq. (2.15b), to produce an expression for the mixed-variable generating function (2.19) for a single integration step.

### 3. Example: quadrupole with octupole component and fringe field

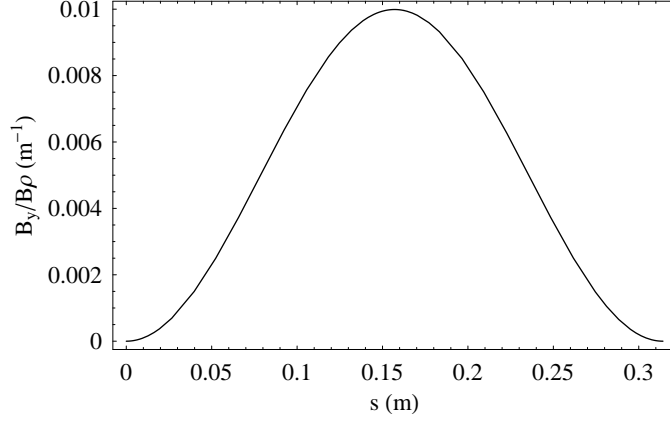
As an example, we illustrate the construction of the map, in the form of a mixed-variable generating function of the second kind, for a quadrupole with strong octupole component. The axis of the magnet (on which the field is zero) defines the reference trajectory. We allow both the quadrupole and octupole components to vary with position along the axis of the magnet, so as to represent some fringe field region at the entrance and exit of the magnet. An appropriate representation of the field, including the longitudinal variation, is given by Dragt’s generalised gradients [13]. The appropriate expressions are given in reference [27]; for convenient reference, they are reproduced in Appendix B. In the notation used there, our field is defined by:

$$C_2(s) = \frac{1}{2} \frac{k_1}{B\rho} \sin^2(k_s s), \quad (3.1)$$

$$C_4(s) = \frac{1}{4!} \frac{k_3}{B\rho} \sin^2(k_s s), \quad (3.2)$$

with  $k_1 = -10 \text{ m}^{-2}$ ,  $k_3 = 6 \times 10^4 \text{ m}^{-4}$ , and  $k_s = 10 \text{ m}^{-1}$ . The total length of the magnet is defined to be  $L = \pi/k_s$ . The expressions for the vector potential involve infinite summations: in practice, these must be truncated at some order in the co-ordinates. Although the fields corresponding to the truncated potentials strictly speaking do not satisfy Maxwell’s equations, the error is of order higher than the truncation; and if the truncation is at significantly high order, the transfer map, itself computed only up to some order, is not affected. Note that the expressions for the transformations in the symplectic integrator involve the vector potential normalised by  $P_0/q$ : therefore, it is not necessary to specify either the particle charge or the reference momentum. The normalised vertical field along the line  $x = -1 \text{ mm}$ ,  $y = 0$  from  $s = 0$  to  $s = L$  is shown in figure 1.

Using the generalised gradients (3.1) and (3.2), we compute the mixed-variable generating function for the transfer map from  $s = 0$  to  $s = L$ , using the fourth-order Runge-Kutta integrator (Section 2.2), and the iterative method for composing the mixed-variable generating functions (Section 2.4). The Hamiltonian is given by Eq. (2.7), with the square root expanded to sixth order



**Figure 1.** Vertical normalised field component in the example magnet, along the line  $x = -1$  mm,  $y = 0$  from  $s = 0$  to  $s = L$ .

in the dynamical variables. Terms up to sixth order in the co-ordinates are retained in the expression for the vector potential. We retain terms to fourteenth order in the dynamical variables when composing the generating function, to reduce feed-down errors. A step size of  $\Delta\sigma = L/1024$  was used in the integration. All computations were performed in Mathematica.

The transfer functions (final co-ordinate and momentum, as functions of initial co-ordinate and momentum) are shown in figure 2. For comparison, we also computed the transfer functions using direct numerical integration of Hamilton’s equations, using the Hamiltonian (2.7) (without any approximation for the square root). The residuals between the transfer functions obtained from the mixed-variable generating function, and the transfer functions obtained from the numerical integration are also shown in figure 2.

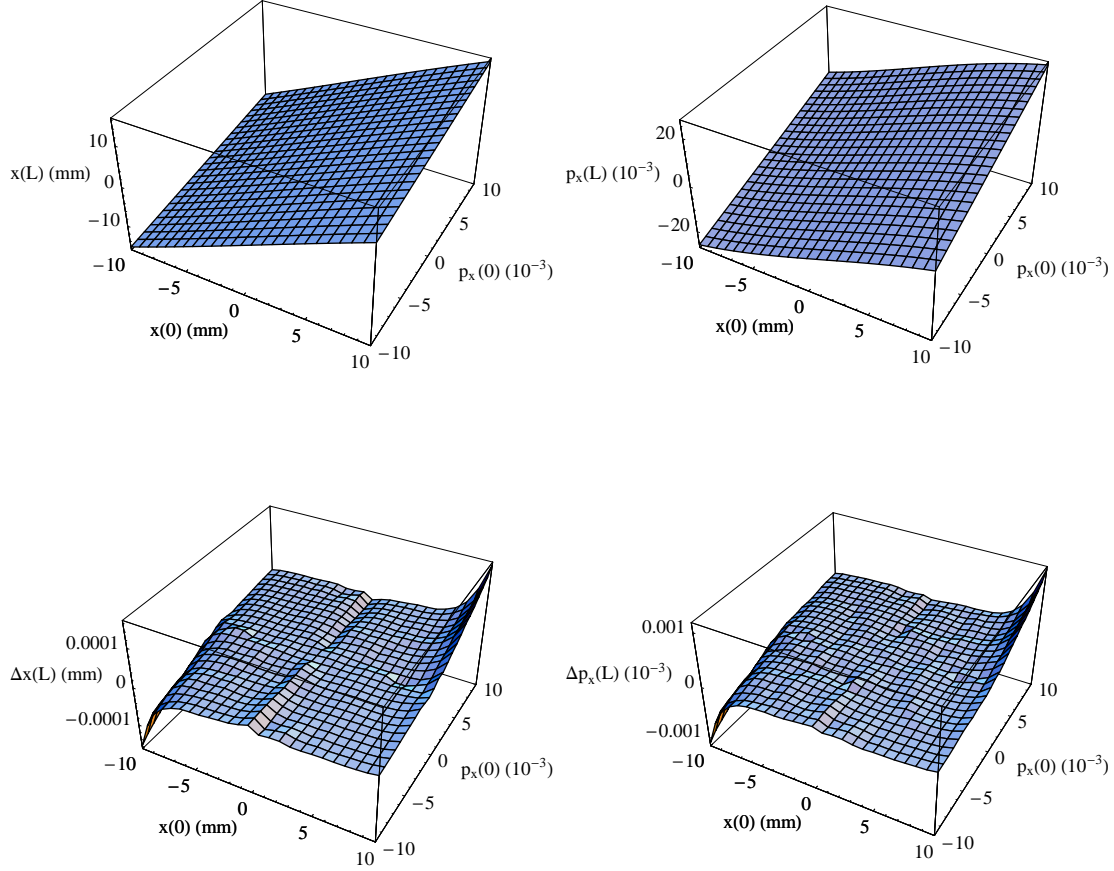
The residuals appear small over the range of variables plotted. To investigate the residuals more closely, we look specifically at the value of  $p_x(L)$  as a function of  $x(0)$ , for  $p_x(0) = 0$ : we refer to this as the 1-D (one-dimensional) transfer function. A plot of the 1-D transfer function is shown in figure 3 (note that there is some third-order curvature evident, from the octupole component in the quadrupole). Also shown in figure 3 is the difference (the residual) between the 1-D transfer functions obtained from the mixed-variable generating function and from direct numerical integration. The residual can be fitted (apart from some fluctuations, which may be due to limits on numerical precision) using a polynomial with only a single term, in  $x^7$ . This indicates that the residual is dominated by the seventh-order terms in the map.

We can also compare the coefficients in the 1-D transfer function (shown in figure 3, left):

$$p_x(L) = \sum_{m=1}^5 h_m x(0)^m. \quad (3.3)$$

The coefficients  $h_m$  can be obtained using various techniques; a comparison of the results from some methods are shown in table 1. First, we obtain the  $h_m$  directly from the generating function, by solving:

$$p_{x1} = 0 = \frac{\partial F}{\partial x_1}, \quad (3.4)$$

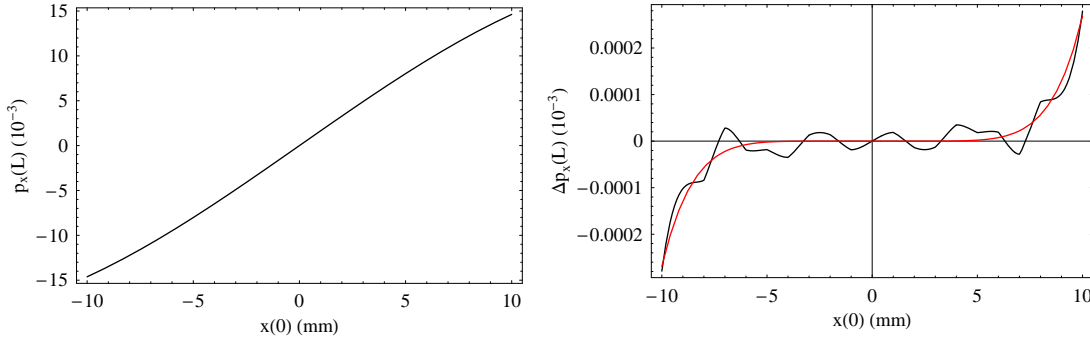


**Figure 2.** Transfer functions in the example magnet. Top: horizontal co-ordinate (left) and normalised momentum (right) at  $s = L$  as functions of co-ordinate and momentum at  $s = 0$ . Bottom: residuals for the final horizontal co-ordinate (left) and normalised momentum (right), between the values calculated using the mixed-variable generating function, and a purely numerical integration through the field.

where  $F$  is the generating function for a map through the entire magnet; the solution may be expanded in a power series in  $x_1$ , which gives the  $h_m$  directly. Second, we can obtain the  $h_m$  from a polynomial fit to the 1-D transfer function obtained by numerical integration. Finally, estimated values for the  $h_m$  may be obtained by integrating the appropriate normalised multipole component along the reference trajectory:

$$h_m \approx -\frac{1}{m!} \int k_m ds. \quad (3.5)$$

The results from the first two methods are in good agreement up to the fourth-order ( $h_4$ ), and in reasonable agreement for  $h_5$ . The values obtained using the third method (integrating the normalised multipole component along the reference trajectory, Eq. (3.5)) are only approximate: effects associated with the fringe fields, and the non-zero length of the magnet, are not properly taken into



**Figure 3.** Left: normalised horizontal momentum at  $s = L$ , as a function of horizontal co-ordinate at  $s = 0$ , for  $p_x(0) = 0$ . Right: residual in final normalised horizontal momentum between values calculated from the mixed-variable generating function and from direct numerical integration. The black line shows the residual, the red line shows a fit to the residual, using a function of the form  $\Delta p_x(L) \propto x(0)^7$ .

**Table 1.** Coefficients in the transfer function, derived from the mixed-variable generating function (MVGf), and by a polynomial fit to the tracking results using direct numerical integration (NI). Also shown for comparison are the values that would be expected from the field integrals (FI).

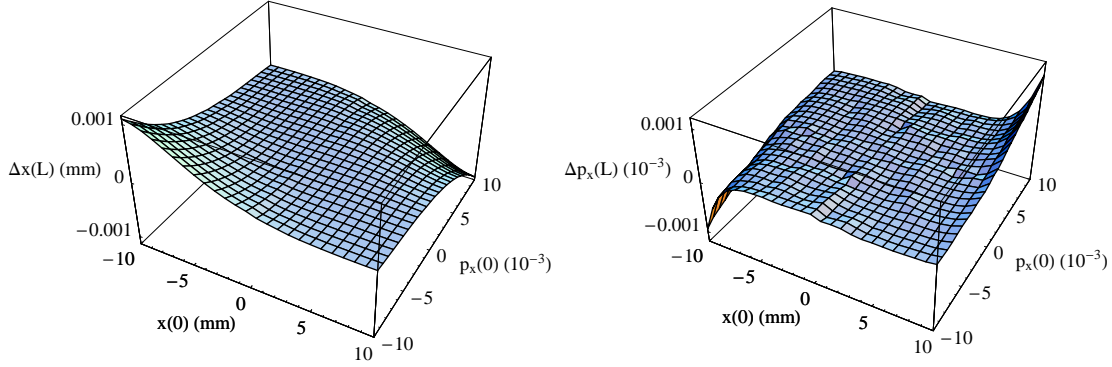
m	$h_m$ MVGf	$h_m$ NI	$h_m$ FI
1	$1.65228 \text{ m}^{-1}$	$1.65226 \text{ m}^{-1}$	$1.5708 \text{ m}^{-1}$
2	$-1.00075 \times 10^{-13} \text{ m}^{-2}$	$-1.00075 \times 10^{-13} \text{ m}^{-2}$	0
3	$-1933.15 \text{ m}^{-3}$	$-1930.82 \text{ m}^{-3}$	$-1570.8 \text{ m}^{-3}$
4	$2.21872 \times 10^{-9} \text{ m}^{-4}$	$2.21798 \times 10^{-9} \text{ m}^{-4}$	0
5	$3.84174 \times 10^5 \text{ m}^{-5}$	$3.30479 \times 10^5 \text{ m}^{-5}$	0

account.

Finally, we can see the effect the paraxial approximation would have if applied to this example, by comparing the residuals between the results of the direct numerical integration using the Hamiltonian (2.7), and the Hamiltonian in the paraxial approximation (2.10). The difference between the two is small in this case, but visible if we compare the difference in the final co-ordinate, as a function of initial co-ordinate and momentum: this is shown in figure 4. The residuals are an order of magnitude larger than those between the direct numerical integration for the “exact” Hamiltonian, and the integration (with mixed-variable generating functions) for the Hamiltonian expanded to sixth order in the dynamical variables. This suggests that using mixed-variable generating functions, we can readily construct an exactly symplectic high-order map to good accuracy, without needing to make the paraxial approximation in the Hamiltonian for the system.

## A. Proof of the inductive formula for solving a system of polynomial equations

We wish to show that (2.33) is a solution to (2.29). Assuming that  $\hat{\mathbf{T}}_1$  is invertible, we multiply



**Figure 4.** Residuals in the final co-ordinate (left) and final momentum (right) as a function of initial co-ordinate and momentum. The residuals are calculated as the difference between the results found from the mixed-variable generating function for the “exact” Hamiltonian, and the numerical integration of the Hamiltonian in the paraxial approximation.

Eq. (2.29) by  $-\hat{\mathbf{T}}_1^{-1}$  to give:

$$\sum_{r=2}^N \mathbf{T}_r(\mathbf{X}, \dots, \mathbf{X}) - \mathbf{X} + \mathbf{T}_0 = O(\varepsilon^{N+1}). \quad (\text{A.1})$$

Let  $\mathbf{S}_n$  be the left hand side of (A.1):

$$\mathbf{S}_n = \sum_{r=2}^N \mathbf{T}_r(\overbrace{\mathbf{X}, \dots, \mathbf{X}}^r) - \mathbf{X} + \mathbf{T}_0. \quad (\text{A.2})$$

Substituting for  $\mathbf{X}$  from Eq. (2.32) gives:

$$\mathbf{S}_n = \sum_{r=2}^N \mathbf{T}_r\left(\sum_{m_1=1}^{\infty} \mathbf{c}_{m_1}, \dots, \sum_{m_r=1}^{\infty} \mathbf{c}_{m_r}\right) - \sum_{r=1}^{\infty} \mathbf{c}_r + \mathbf{T}_0. \quad (\text{A.3})$$

Since  $\mathbf{c}_1 = \mathbf{T}_0$ , we find:

$$\mathbf{S}_n = \sum_{r=2}^N \mathbf{T}_r\left(\sum_{m_1=1}^N \mathbf{c}_{m_1}, \dots, \sum_{m_r=1}^N \mathbf{c}_{m_r}\right) - \sum_{r=2}^N \mathbf{c}_r + O(\varepsilon^{N+1}) \quad (\text{A.4})$$

$$= \sum_{r=2}^N \sum_{m_1=1}^N \dots \sum_{m_r=1}^N \mathbf{T}_r(\mathbf{c}_{m_1}, \dots, \mathbf{c}_{m_r}) - \sum_{r=2}^N \mathbf{c}_r + O(\varepsilon^{N+1}) \quad (\text{A.5})$$

$$= \sum_{r=2}^N \sum_{k=2}^N \sum_{\substack{m_1, \dots, m_k=1 \\ m_1 + \dots + m_k = k}}^N \mathbf{T}_r(\mathbf{c}_{m_1}, \dots, \mathbf{c}_{m_r}) - \sum_{k=2}^N \mathbf{c}_k + O(\varepsilon^{N+1}) \quad (\text{A.6})$$

$$= \sum_{k=2}^N \left( -\mathbf{c}_k + \sum_{r=2}^N \sum_{\substack{m_1, \dots, m_k=1 \\ m_1 + \dots + m_k = k}}^N \mathbf{T}_r(\mathbf{c}_{m_1}, \dots, \mathbf{c}_{m_r}) \right) + O(\varepsilon^{N+1}). \quad (\text{A.7})$$

For  $r > k$  the set  $\{(m_1, \dots, m_r) \mid m_i \geq 1 \text{ and } m_1 + \dots + m_r = k\} = \emptyset$ , hence:

$$\mathbf{S}_n = \sum_{k=2}^N \left( -\mathbf{c}_k + \sum_{r=2}^k \sum_{\substack{m_1, \dots, m_k=1 \\ m_1 + \dots + m_k = k}} (\mathbf{c}_{I_1}, \dots, \mathbf{c}_{I_n}) \right) + O(\epsilon^{N+1}). \quad (\text{A.8})$$

Substituting for  $\mathbf{c}_k$  from Eq. (2.33) gives:

$$\mathbf{S}_n = O(\epsilon^{N+1}), \quad (\text{A.9})$$

which is Eq. (2.29).

## B. Generalised gradients

In Cartesian variables, a vector potential in the Coulomb gauge can be expressed as [27]:

$$A_x = \frac{1}{2} \sum_{m=1}^{\infty} \Re(x + iy)^{m+1} \sum_{\ell=0}^{\infty} (-1)^{\ell} \frac{m!}{2^{2\ell} \ell! (\ell + m + 1)!} C_m^{[2\ell+1]}(z) (x^2 + y^2)^{\ell}, \quad (\text{B.1})$$

$$A_y = \frac{1}{2} \sum_{m=1}^{\infty} \Im(x + iy)^{m+1} \sum_{\ell=0}^{\infty} (-1)^{\ell} \frac{m!}{2^{2\ell} \ell! (\ell + m + 1)!} C_m^{[2\ell+1]}(z) (x^2 + y^2)^{\ell}, \quad (\text{B.2})$$

$$A_z = - \sum_{m=1}^{\infty} \Re(x + iy)^m \sum_{\ell=0}^{\infty} (-1)^{\ell} \frac{m!}{2^{2\ell} \ell! (\ell + m)!} C_m^{[2\ell]}(z) (x^2 + y^2)^{\ell}, \quad (\text{B.3})$$

where:

$$C_m^{[k]} = \frac{d^k C_m}{dz^k}. \quad (\text{B.4})$$

Maxwell's equations in free space are satisfied for any differentiable functions  $C_m(z)$ , which are called the generalised gradients. The conventional multipole expansion is obtained in the case that the  $C_m(z)$  are independent of  $z$ ; then,  $C_1$  represents the dipole field component,  $C_2$  the quadrupole field component, and so on.

The symplectic integrator described in Section 2 uses a vector potential in the gauge:

$$A_y = 0. \quad (\text{B.5})$$

This is readily obtained from the above expressions using the gauge transformation:

$$\vec{A}' = \vec{A} - \nabla \psi, \quad (\text{B.6})$$

where:

$$\psi = \int A_y dy. \quad (\text{B.7})$$

## Acknowledgments

We should like to thank B. Muratori for suggesting the use of symplectic Runge-Kutta integration for particle tracking. This work was supported by the Science and Technology Facilities Council, UK.



## References

- [1] K.L. Brown, *A first- and second-order matrix theory for the design of beam transport systems and charged particle spectrometers*, SLAC Report-75 (Revision 4), (1982).
- [2] E. Forest, *Beam Dynamics: A New Attitude and Framework*, Harwood Academic Publishers, Amsterdam 1998.
- [3] M. Berz, *Modern Map Methods in Particle Beam Physics*, Academic Press, San Diego 1999.
- [4] E. Forest, *Geometric integration for particle accelerators*, *J. Phys. A: Math. Gen.* **39** (2006) 5321–5377.
- [5] A.J. Dragt, *Lie Methods for Nonlinear Dynamics with Applications to Accelerator Physics*, [www.physics.umd.edu/dsat](http://www.physics.umd.edu/dsat) (December 2011).
- [6] M. Berz, *Differential Algebraic Techniques*, in A.W. Chao and M. Tigner (editors), *Handbook of Accelerator Physics and Engineering*, World Scientific, Singapore 1999, pp. 82–85.
- [7] Y. Yan, P. Channell, M. Syphers, *An algorithm for symplectic implicit Taylor-map tracking*, *SSCL-Preprint-157* (1992).
- [8] R. Kleiss, F. Schmidt, Y. Yan, F. Zimmermann, *On the feasibility of tracking with differential-algebra maps in long-term stability studies for large hadron colliders*, *DESY-HERA-92-01*, *CERN-SL-92-02-AP*, *SSCL-564* (1992).
- [9] J.S. Berg, R.L. Warnock, R.D. Ruth, E. Forest, *Construction of symplectic maps for nonlinear motion of particles in accelerators*, *Phys. Rev. E* **49** (1994) 722–739.
- [10] D.T. Abell, *Analytic properties and Cremona approximation of transfer maps for Hamiltonian systems*, *Doctoral Dissertation*, University of Maryland (1995).
- [11] D.T. Abell, E. McIntosh, F. Schmidt, *Fast symplectic map tracking for the CERN Large Hadron Collider*, *Phys. Rev. ST Accel. Beams* **6**, 064001 (2003).
- [12] A.J. Dragt, *Lie Maps*, in A.W. Chao and M. Tigner (editors), *Handbook of Accelerator Physics and Engineering*, World Scientific, Singapore 1999, pp. 76–82.
- [13] M. Venturini, A.J. Dragt, *Accurate computation of transfer maps from magnetic field data*, *Nucl. Instrum. Methods Phys. Res. Sect. A* **427** (1999) 387–392.
- [14] H. Goldstein, *Classical Mechanics*, Addison-Wesley, Reading MA 1980 (Second Edition).
- [15] M. Berz, *Symplectic tracking through circular accelerators with high order maps*, in W. Scandale and G. Turchetti (editors), *Nonlinear Problems in Future Particle Accelerators*, World Scientific, Singapore 1991.
- [16] Y.T. Yan, P.J. Channell, M. Li, and M.J. Syphers, *Long term tracking with symplectic implicit one-turn maps*, in proceedings of the *1993 Particle Accelerator Conference*, May 17–20, 1993, Washington, DC, USA, pp. 38–40.
- [17] J.S. Berg, R.L. Warnock, R.D. Ruth, E. Forest, *Full-turn symplectic map from a generator in a fourier-spline basis*, *SLAC-PUB-6164* (1993).
- [18] R.L. Warnock, J.S. Berg, *Fast symplectic mapping and long-term stability near broad resonances*, *SLAC-PUB-7464* (1997).
- [19] B. Erdélyi, *Global theory of extended generating functions*, (unpublished) [nicadd.niu.edu/research/beams/erdelyimath.pdf](http://nicadd.niu.edu/research/beams/erdelyimath.pdf) (2006).

- [20] B. Erdélyi, M. Berz, *Optimal symplectic approximation of Hamiltonian flows*, *Phys. Rev. Lett.* **87**, 114302 (2001).
- [21] B. Erdélyi, M. Berz, *Local theory and applications of extended generating functions*, *Int. J. Pure and Applied Math.* **11** (2004) 241–280.
- [22] Y. Giboudot, *Study of beam dynamics in ns-FFAG EMMA with dynamical map*, *PhD Thesis*, Brunel University (2011).
- [23] Y. Giboudot, A. Wolski, *Use of transfer maps for modelling beam dynamics in a non-scaling fixed-field alternating-gradient accelerator*, submitted to *Phys. Rev. ST Accel. Beams* (2012).
- [24] Y.K. Wu, E. Forest, D.S. Robin, *Explicit symplectic integrator for s-dependent static magnetic field*, *Phys. Rev. E* **68**, 046502 (2003).
- [25] E. Hairer, C. Lubich, G. Wanner, *Geometric Numerical Integration*, Springer, Heidelberg 2006 (Second Edition).
- [26] H. Yoshida, *Numerical Integration Methods*, in A.W. Chao and M. Tigner (editors), *Handbook of Accelerator Physics and Engineering*, World Scientific, Singapore (1999), pp. 85–87.
- [27] Ibid. [5], p. 965.